

AD-A154 097

STATISTICAL ESTIMATION OF SOFTWARE RELIABILITY(U)
CALIFORNIA UNIV BERKELEY OPERATIONS RESEARCH CENTER
S M ROSS MAR 85 ORC-85-3 AFOSR-81-0122

1/1

UNCLASSIFIED

F/G 9/2

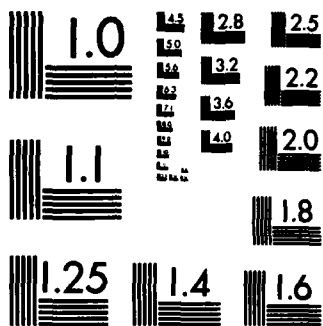
NL

43

42

END

9711



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

2

AD-A154 097

OPERATIONS RESEARCH CENTER

STATISTICAL ESTIMATION OF SOFTWARE RELIABILITY

by

Sheldon M. Ross[†]

✓
ORC-85-3

March 1985

DTIC FILE COPY

DTIC
ELECTE
MAY 24 1985
S B D

UNIVERSITY OF CALIFORNIA



BERKELEY

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

85 04 30 051

STATISTICAL ESTIMATION OF SOFTWARE RELIABILITY

by

Sheldon M. Ross[†]

✓
ORC-85-3

March 1985

DTIC
ELECTE
S MAY 24 1985 D
B

[†]Department of Industrial Engineering and Operations Research,
University of California, Berkeley, California.

This research was supported by the Air Force Office of Scientific
Research (AFSC), USAF, under Grant AFOSR-81-0122 with the University
of California. Reproduction in whole or in part is permitted for
any purpose of the United States Government.

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

AD-A154097

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED			1b. RESTRICTIVE MARKINGS														
2a. SECURITY CLASSIFICATION AUTHORITY			3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited.														
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE																	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) ORC 85-3			5. MONITORING ORGANIZATION REPORT NUMBER(S)														
6a. NAME OF PERFORMING ORGANIZATION Operations Research Center		6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION Air Force Office of Scientific Research														
6c. ADDRESS (City, State and ZIP Code) University of California Berkeley, California 94720			7b. ADDRESS (City, State and ZIP Code) Directorate of Mathematical & Information Sciences, Bolling AFB DC 20332-6448														
8a. NAME OF FUNDING/SPONSORING ORGANIZATION AFOSR		8b. OFFICE SYMBOL (If applicable) NM	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER AFOSR-81-0122														
8c. ADDRESS (City, State and ZIP Code) Bolling AFB DC 20332-6448			10. SOURCE OF FUNDING NOS. <table border="1"><tr><td>PROGRAM ELEMENT NO.</td><td>PROJECT NO.</td><td>TASK NO.</td><td>WORK UNIT NO.</td></tr><tr><td>61102F</td><td>2304</td><td></td><td></td></tr></table>			PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.	61102F	2304						
PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.														
61102F	2304																
11. TITLE (Include Security Classification) STATISTICAL ESTIMATION OF SOFTWARE RELIABILITY																	
12. PERSONAL AUTHOR(S) Sheldon M. Ross																	
13a. TYPE OF REPORT Technical Report		13b. TIME COVERED FROM _____ TO _____		14. DATE OF REPORT (Yr., Mo., Day)													
				15. PAGE COUNT 23													
16. SUPPLEMENTARY NOTATION																	
17. COSATI CODES <table border="1"><tr><td>FIELD</td><td>GROUP</td><td>SUB. GR.</td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>			FIELD	GROUP	SUB. GR.										18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) Keywords: software reliability, error rates, exponential occurrence times		
FIELD	GROUP	SUB. GR.															
19. ABSTRACT (Continue on reverse if necessary and identify by block number) (See Abstract)																	
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED <input checked="" type="checkbox"/> SAME AS RPT. <input type="checkbox"/> DTIC USERS <input type="checkbox"/>			21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED														
22a. NAME OF RESPONSIBLE INDIVIDUAL Major Brian W. Woodruff			22b. TELEPHONE NUMBER (Include Area Code) (202) 767-5027		22c. OFFICE SYMBOL NM												

Abstract

When a new computer software package is developed, a testing procedure is often put into effect to eliminate the faults, or bugs, in the package. One common procedure is to try the package on a set of well known problems to try to see if any errors result. This goes on for some fixed time with all detected errors being noted. Then the testing stops and the package is carefully checked to determine the specific bugs that were responsible for the observed errors, and the package is then altered to remove these bugs. A problem of great importance is the estimation of the error rate of this revised software package.

To model the above, we suppose that initially the package contains m , an unknown number, of bugs which cause errors to occur in accordance with independent Poisson process having unknown rates λ_i , $i = 1, \dots, m$. We suppose that the package is to be run for t time units and that each error is, independently, detected with some known probability p . At the end of this time, a careful check of the package is made to determine the specific bugs that caused the detected errors (that is, a *debugging* takes place). These bugs are then removed and the problem of interest is to determine the error rate for the revised package. In this paper we show how to estimate this quantity under a variety of assumptions as to what is learned when the debugging occur.

Accession For	
NTIS GRA&I	<input checked="checked" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	



STATISTICAL ESTIMATION OF SOFTWARE RELIABILITY

by

Sheldon M. Ross
University of California, Berkeley

0. INTRODUCTION

When a new computer software package is developed, a testing procedure is often put into effect to eliminate the faults, or bugs, in the package. One common procedure is to try the package on a set of well known problems to try to see if any errors result. This goes on for some fixed time with all detected errors being noted. Then the testing stops and the package is carefully checked to determine the specific bugs that were responsible for the observed errors, and the package is then altered to remove these bugs. However, as we cannot be certain that all the bugs in the package have been eliminated, a problem of great importance is the estimation of the error rate of the revised software package.

To model the above, let us suppose that initially the package contains m , an unknown number, of bugs which we will refer to as bug 1, bug 2, ..., bug m . Suppose also that bug i will cause errors to occur in accordance with a Poisson process having an unknown rate λ_i , $i = 1, \dots, m$. Then, for instance, the number of errors due to bug i that occur in any s units of operating time is Poisson distributed with mean $\lambda_i s$. Also suppose that these Poisson processes caused by bugs i , $i = 1, \dots, m$ are independent. Also we suppose that the package is to be run for t time units and we suppose that each error is, independently, detected with some known probability p . At the end of this time, a careful check of the package is made to determine the specific bugs that caused the detected

errors (that is, a *debugging* takes place). These bugs are then removed and the problem of interest is to determine the error rate for the revised package.

The above problem is considered in Section 1 and a preliminary estimation is presented. In Section 2, we make the added assumption that once a given bug has been found, its error rate becomes known. Under this assumption, we show how to improve upon the estimator of Section 1. We also present, in Section 2, an estimator different than that in Section 1 which can be used when error rates are not learned and one that can be used when debuggings necessarily occur whenever an error is detected. In Section 3, we consider the situation where a debugging occurs whenever an error is detected but it need not be successful. In Section 4, we start with a Bayesian model which initially assumes that the number of errors is Poisson distributed with known mean c , and given the number of bugs the failure rates of the bugs are independent with a common known distribution G . We then successively allow, in Section 4.1, the Poisson parameter c to be unknown, and, in Section 4.2, both c and G to be unknown. In both these latter cases, we suppose that a bug's failure rate becomes known when the bug is detected. Interestingly, our estimate when both c and G are unknown is identical to the one given in Section 2. In Section 5, we show how the data at time t can be used to estimate what the total error rate would be at time $t + s$ if the testing were to continue for an additional time s and also present an estimator for the expected number of new bugs that are discovered in $(t, t + s)$.

For a survey of other statistical procedures used in software reliability estimation, the interested reader should see Ramamoorthy and Bastani [7].

1. A PRELIMINARY MODEL

Let

$$\psi_i(t) = \begin{cases} 1 & \text{if bug } i \text{ has not caused a detected error by } t \\ 0 & \text{otherwise .} \end{cases}$$

The quantity we wish to estimate is thus

$$\Lambda(t) = \sum_i \lambda_i \psi_i(t)$$

the error rate of the final package. To start note that

$$\begin{aligned} E[\Lambda(t)] &= \sum_i \lambda_i E[\psi_i(t)] \\ (1) \qquad &= \sum_i \lambda_i e^{-\lambda_i p t} . \end{aligned}$$

Now each of the bugs that are discovered would have been responsible for a certain number of detected errors. Let us denote by $M_j(t)$ the number of bugs that were responsible for j detected errors, $j \geq 1$. That is, $M_1(t)$ is the number of bugs that caused exactly 1 detected error, $M_2(t)$ is the number that caused 2 errors, and so on, with $\sum_j j M_j(t)$ equalling the total number of detected errors. To compute $E[M_1(t)]$, let us define the indicator variables, $I_i(t)$, $i \geq 1$, by

$$I_i(t) = \begin{cases} 1 & \text{bug } i \text{ causes exactly 1 detected error} \\ 0 & \text{otherwise .} \end{cases}$$

Then,

$$M_1(t) = \sum_i I_i(t)$$

and so

$$(2) \quad E[M_1(t)] = \sum_i E[I_i(t)] = \sum_i \lambda_i p t e^{-\lambda_i p t}.$$

Thus, from (1) and (2) we obtain the intriguing result that

$$E\left[\Lambda(t) - \frac{M_1(t)}{pt}\right] = 0.$$

This suggests the possible use of $\frac{M_1(t)}{pt}$ as an estimate of $\Lambda(t)$. To determine whether or not $\frac{M_1(t)}{pt}$ constitutes a "good" estimate of $\Lambda(t)$ we shall look at how far apart these 2 quantities tend to be. That is, we will compute

$$\begin{aligned} E\left[\left(\Lambda(t) - \frac{M_1(t)}{pt}\right)^2\right] &= \text{Var}\left(\Lambda(t) - \frac{M_1(t)}{pt}\right) \\ &= \text{Var}(\Lambda(t)) - \frac{2}{pt} \text{Cov}(\Lambda(t), M_1(t)) + \frac{1}{p^2 t^2} \text{Var}(M_1(t)). \end{aligned}$$

Now

$$\text{Var}(\Lambda(t)) = \sum_i \lambda_i^2 \text{Var}(\psi_i(t)) = \sum_i \lambda_i^2 e^{-\lambda_i p t} \left(1 - e^{-\lambda_i p t}\right)$$

$$\text{Var}(M_1(t)) = \sum_i \text{Var}(I_i(t)) = \sum_i \lambda_i p t e^{-\lambda_i p t} \left(1 - \lambda_i p t e^{-\lambda_i p t}\right)$$

$$\begin{aligned}
\text{Cov}(\Lambda(t), M_1(t)) &= \text{Cov}\left(\sum_i \lambda_i \psi_i(t), \sum_j I_j(t)\right) \\
&= \sum_i \sum_j \text{Cov}(\lambda_i \psi_i(t), I_j(t)) \\
&= \sum_i \lambda_i \text{Cov}(\psi_i(t), I_i(t)) \\
&= - \sum_i \lambda_i e^{-\lambda_i p t} \lambda_i p t e^{-\lambda_i p t}
\end{aligned}$$

where the last two equalities follow since $\psi_i(t)$ and $I_j(t)$ are independent when $i \neq j$ as they refer to different Poisson processes, and $\psi_i(t)I_i(t) = 0$. Hence we obtain that

$$\begin{aligned}
E\left[\left(\Lambda(t) - \frac{M_1(t)}{pt}\right)^2\right] &= \sum_i \lambda_i^2 e^{-\lambda_i p t} + \frac{1}{pt} \sum_i \lambda_i e^{-\lambda_i p t} \\
&= \frac{E[M_1(t) + 2M_2(t)]}{p^2 t^2}
\end{aligned}$$

where the last equality follows since

$$E[M_2(t)] = \sum_i e^{-\lambda_i p t} (\lambda_i p t)^2 / 2.$$

Thus we can estimate the average square of the difference between $\Lambda(t)$ and $\frac{M_1(t)}{pt}$ by the observed value of $M_1(t) + 2M_2(t)$ divided by $p^2 t^2$.

Remark:

The above analysis is similar in spirit to that done in Robbins [4]. Very similar results have also been presented by Diaconis in a set of unpublished notes on decision theory.

2. ERROR RATES LEARNED UPON DEBUGGING

Let us now suppose that the failure rate due to a bug becomes known once the bug has been discovered. That is, we are supposing that based on our experience we are able to accurately estimate the failure rate due to any particular bug once it has been discovered.

Let R denote the number of bugs discovered by time t and let $\Lambda_1, \dots, \Lambda_R$ be their failure rates. Then

$$E[M_1(t) \mid R, \Lambda_1, \dots, \Lambda_R] = \sum_i \frac{\lambda_i p t e^{-\lambda_i p t}}{1 - e^{-\lambda_i p t}} (1 - \psi_i(t)) .$$

We shall consider $\frac{1}{pt} E[M_1(t) \mid R, \Lambda_1, \dots, \Lambda_R]$ as an estimate of $\Lambda(t)$.

Since

$$E[E[M_1(t) \mid R, \Lambda_1, \dots, \Lambda_R]] = E[M_1(t)] = ptE[\Lambda(t)]$$

its square error loss is as follows:

$$\begin{aligned} & \text{Var} \left(\sum_i \frac{\lambda_i e^{-\lambda_i p t}}{1 - e^{-\lambda_i p t}} (1 - \psi_i(t)) - \Lambda(t) \right) \\ &= \text{Var} \left(\sum_i \left(\frac{\lambda_i e^{-\lambda_i p t}}{1 - e^{-\lambda_i p t}} + \lambda_i \right) \psi_i(t) \right) \\ &= \sum_i \left(\frac{\lambda_i}{1 - e^{-\lambda_i p t}} \right)^2 e^{-\lambda_i p t} (1 - e^{-\lambda_i p t}) \\ &= \sum_i \frac{\lambda_i^2 e^{-\lambda_i p t}}{1 - e^{-\lambda_i p t}} . \end{aligned}$$

It should be noted that as

$$\frac{\lambda^2}{1 - e^{-\lambda pt}} \leq \lambda^2 + \frac{\lambda}{pt}, \quad \lambda \geq 0, \quad t \geq 0$$

it follows that

$$\sum_i \frac{\lambda_i e^{-\lambda_i pt}}{1 - e^{-\lambda_i pt}} (1 - \psi_i(t)) = \sum_{i=1}^R \frac{\Lambda_i e^{-\Lambda_i pt}}{1 - e^{-\Lambda_i pt}}$$

is a better estimator of $\Lambda(t)$ than is $M_1(t)/tp$, and should thus be preferred whenever the finding of a bug also reveals its failure rate.

Remarks:

- (i) The above estimator can be used whether debugging is performed whenever an error is detected and the bug removed or if the debugging is performed at time t . In this latter case, another estimator is obtained by first noting that, independent of the bugs causing the errors, detected errors will occur at a Poisson rate $p \sum_i \lambda_i$. Hence, letting $D(t) = \sum_i iM_i(t)$ denote the number of detected errors by t , then $\frac{D(t)}{pt}$ can be used to estimate $\sum_i \lambda_i$. As $\sum_{i=1}^R \Lambda_i$ is the error rate due to those bugs that have caused detected errors, it thus follows that $\frac{D(t)}{pt} - \sum_{i=1}^R \Lambda_i$ is an estimator of $\Lambda(t)$. To evaluate its worth, note that

$$E \left[\sum_{i=1}^R \Lambda_i \right] = E \left[\sum_i \lambda_i (1 - \psi_i(t)) \right] = \sum_i \lambda_i (1 - e^{-\lambda_i pt})$$

and so

$$E \left[\frac{D(t)}{pt} - \sum_{i=1}^R \Lambda_i \right] = \sum \lambda_i e^{-\lambda_i tp} = E[\Lambda(t)] .$$

Also

$$\text{Var} \left(\frac{D(t)}{pt} - \sum_{i=1}^R \Lambda_i - \Lambda(t) \right) = \text{Var} \left(\frac{D(t)}{pt} \right) = \frac{\sum \lambda_i}{pt} .$$

However,

$$\sum_i \frac{\lambda_i}{tp} = \frac{E[D(t)]}{p^2 t^2} \geq \frac{E[M_1(t) + 2M_2(t)]}{p^2 t^2} = \text{Var} \left(\Lambda(t) - \frac{M_1(t)}{pt} \right)$$

it follows that this estimator is not even as good as $\frac{M_1(t)}{pt}$.

[Of course, the estimator could be improved somewhat by considering

$$\max \left(0, \frac{D(t)}{pt} - \sum_{i=1}^R \Lambda_i \right) .$$

- (ii) If we are not willing to suppose that we can accurately estimate the failure rate of a discovered bug, another approach is to suppose that you can express your feelings in terms of a probability distribution G_i on Λ_i . This suggests the estimator

$$\sum_{i=1}^R E_{G_i} \left[\frac{\Lambda_i e^{-\Lambda_i tp}}{1 - e^{-\Lambda_i tp}} \mid \text{data} \right]$$

where G_i is one's feelings about the i th discovered bug (after it has been identified). If the debugging is performed at time t , then the data consists of the number of detected errors due to the i th bug to be discovered.

- (iii) If one is not willing to take the above "Bayesian approach", one could try maximum likelihood estimates. For instance, if all debugging is performed at time t , then with $N_i(t)$ denoting the number of detected errors caused by bug i , the estimator

$$E_3(t) = \frac{1}{pt} \sum_i \frac{N_i(t) e^{-N_i(t)}}{1 - e^{-N_i(t)}} = \frac{1}{pt} \sum_i M_i(t) \frac{ie^{-i}}{1 - e^{-i}}$$

is suggested. Whereas additional numerical work is needed to see how this estimator compares with $\frac{M_1(t)}{pt}$, preliminary simulation investigations show that it compares quite favorably (see Table 1).

- (iv) Suppose that the nature of the problem is such that a debugging must take place whenever an error is detected and the bug removed. If we are able to determine the bug's failure rate, then this case reduces to one originally considered in this section. However, let us now suppose that the bugs discovery sheds no light on its failure rate. If T_i , $i = 1, \dots, R$ denote the times at which the detected errors occur, then the MLE of $\Lambda_1, \dots, \Lambda_R$ is $1/T_1, \dots, 1/T_R$. Hence a natural estimator to consider in this case is

$$E_4(t) = \sum_{i=1}^R \frac{e^{-pt/T_i}}{T_i(1 - e^{-pt/T_i})} = \frac{1}{t} \sum_{i=1}^R \frac{e^{-p/\alpha_i}}{\alpha_i(1 - e^{-p/\alpha_i})}$$

where $\alpha_i = T_i/t$.

- (v) The following is a partial summary of extensive simulations.

TABLE 1

Summary of Simulation Results: Average values based on 100 simulations: $p = t = 1$

$$E_1 = M_1, E_2 = \sum_{i=1}^R \frac{\lambda_i e^{-\lambda_i}}{1 - e^{-\lambda_i}}$$

$$E_3 = \sum_j M_j \frac{j e^{-j}}{1 - e^{-j}}, E_4 = \sum_{i=1}^R \frac{e^{-1/T_i}}{T_i (1 - e^{-1/T_i})}$$

Average values: based on 100 simulations

		$(E_1 - \lambda(1))^2$	$(E_2 - \lambda(1))^2$	$(E_3 - \lambda(1))^2$	$(E_4 - \lambda(1))^2$
50 bugs	unif (0,3)				
	Case 1	35.364	23.753	26.155	60.282
	Case 2	38.678	27.279	27.559	68.528
50 bugs	unif (0,4)				
	Case 1	30.311	23.772	25.027	53.324
	Case 2	36.569	25.814	26.169	64.773
50 bugs	unif (0,5)				
	Case 1	19.881	15.914	16.593	27.751
	Case 2	30.431	27.723	27.061	40.583
	Case 3	36.133	25.341	26.418	48.156
	Case 4	29.588	22.255	24.368	25.996
	Case 5	30.693	25.802	27.712	34.181
	Case 6	23.772	18.534	21.588	22.915
50 bugs	unif (0,6)				
	Case 1	21.528	15.989	17.998	18.658
	Case 2	21.284	18.854	17.462	29.050
	Case 3	25.738	21.613	21.447	30.873
	Case 4	21.306	20.769	18.973	24.786
50 bugs	unif (0,7)				
	Case 1	33.928	25.449	26.865	34.641
	Case 2	28.415	21.366	22.286	29.503

(vi) The results of this section are consistent with the hypothesis that in attempting to estimate $\Lambda(t)$ the best one can do is to estimate $E[\Lambda(t)]$. That is, suppose one somehow knows the value $\sum_i \lambda_i e^{-\lambda_i p t}$. Then the author suggests that the data should be ignored and $\Lambda(t)$ should be estimated by $\sum_i \lambda_i e^{-\lambda_i p t}$. As

$$\begin{aligned} \text{Var}(\Lambda(t)) &= \sum_i \lambda_i^2 e^{-\lambda_i t p} (1 - e^{-\lambda_i t p}) \\ &< \sum_i \frac{\lambda_i^2 e^{-\lambda_i t p}}{1 - e^{-\lambda_i t p}} \end{aligned}$$

it follows that $E[\Lambda(t)]$ is a better estimate of $\Lambda(t)$ than is $\sum_{i=1}^R \frac{\lambda_i e^{-\lambda_i t p}}{1 - e^{-\lambda_i t p}}$. Of course $E[\Lambda(t)]$ is unknown and cannot be directly implemented as an estimator. (The results of Section 4 also indicates that, in effect, we are really trying to estimate $E[\Lambda(t)]$ and not $\Lambda(t)$).

3. ALLOWING FOR UNSUCCESSFUL DEBUGGINGS

Suppose as in the previous section, that detected errors lead to immediate debugging with the failure rate of the responsible bug being determined. However, let us now suppose that the debugging is only successful with probability α . That is, with probability $1 - \alpha$ a new bug, which we will suppose has the same failure rate as the bug just removed, is created. (Thus we can think of the newly created bug as either being the old bug which was not successfully eliminated or as being a brand new bug caused by our change in the program that eliminated the old bug and which has the same failure rate as the old bug). Suppose also that when a debugging takes place, we are able to tell whether the responsible bug was initially present or was created by a previous debugging.

We can estimate the failure rate at time t as follows: Let us start by adopting the interpretation that if a debugging is unsuccessful, then the responsible bug remains in the program. That is, we are identifying any newly created bug as being identical to the one removed. Suppose that R distinct bugs--having failure rates $\lambda_1, \dots, \lambda_R$ --have been discovered, with $L_i, i = 1, \dots, R$ representing the last time that the bug with rate λ_i

has been responsible for a detected error. Then $\sum_{i=1}^R \frac{\lambda_i e^{-\lambda_i t p}}{1 - e^{-\lambda_i t p}}$ estimates

the error rate at time t of those bugs that have not yet appeared. Also the discovered bug with rate λ_i will still be present in the package at

time t with probability $\frac{e^{-\lambda_i(t-L_i)}(1-\alpha)}{\alpha + (1-\alpha)e^{-\lambda_i(t-L_i)}}$. Hence the total rate at

time t can be estimated from

$$\sum_{i=1}^R \lambda_i \left[\frac{e^{-\lambda_i t p}}{1 - e^{-\lambda_i t p}} + \frac{1 - \alpha}{1 - \alpha + \alpha e^{-\lambda_i(t-L_i)}} \right].$$

4. A BAYESIAN MODEL

To formulate a Bayesian model, we need specify a prior joint distribution for $m, \lambda_1, \dots, \lambda_m$. As there are a large number of possible things that could go wrong when putting together a software package, each having a small probability of going wrong, it seems reasonable to suppose that m , the number of bugs, has a Poisson distribution. Also, given m , we shall suppose that the resulting failure rates are independent and identically distributed. So let us make the following assumption.

Assumption:

The number of bugs m has a Poisson distribution with mean c ; and given $m, \lambda_1, \dots, \lambda_m$ are independent and have the common distribution G . Both c and G are assumed to be known.

We shall assume that once a bug is detected, its failure rate becomes known and the bug is eliminated. (That is, α of Section 3 is taken to equal 1).

As each of the Poisson number of bugs will independently result in a detected error with probability given by

$$P\{\text{bug has a detected error}\} = \int (1 - e^{-\lambda p t}) dG(\lambda)$$

it follows that the number of discovered bugs is Poisson with mean $c \int (1 - e^{-\lambda p t}) dG(\lambda)$ and is independent of the number of undetected bugs which is Poisson with mean $c \int e^{-\lambda p t} dG(\lambda)$. Also the conditional distribution of a bugs failure rate, given that the bug is not discovered is as follows:

$$dG(\lambda \mid \text{not discovered}) = \frac{e^{-\lambda p t} dG(\lambda)}{\int e^{-\lambda p t} dG(\lambda)}.$$

Hence,

$$E[\lambda \mid \text{not discovered}] = \frac{\int \lambda e^{-\lambda p t} dG(\lambda)}{\int e^{-\lambda p t} dG(\lambda)}$$

and thus

$$\begin{aligned} E[\Lambda(t) \mid \text{data}] &= c \int e^{-\lambda p t} dG(\lambda) E[\lambda \mid \text{not discovered}] \\ &= c \int \lambda e^{-\lambda p t} dG(\lambda) . \end{aligned}$$

That is, the Bayes estimator with respect to square error loss is independent of the data and is as given above.

4.1 Unknown c

If c is unknown in the above model, then we can estimate it by

$\frac{R}{\int (1 - e^{-\lambda p t}) dG(\lambda)}$, where R is the number of discovered bugs (and thus has a Poisson distribution with mean $c \int (1 - e^{-\lambda p t}) dG(\lambda)$). Hence, we can estimate the Bayes estimator as follows:

$$E[\Lambda(t) \mid \text{data}] \stackrel{\text{est}}{=} \frac{R \int \lambda e^{-\lambda p t} dG(\lambda)}{\int (1 - e^{-\lambda p t}) dG(\lambda)} .$$

4.2 Both c And G Unknown

Note first that the conditional distribution of the fault rate of a bug that is discovered is as follows:

$$dG(\lambda \mid \text{discovered}) = \frac{(1 - e^{-\lambda p t}) dG(\lambda)}{\int (1 - e^{-\lambda p t}) dG(\lambda)} .$$

Hence,

$$E \left[\frac{\lambda e^{-\lambda tp}}{1 - e^{-\lambda tp}} \mid \text{discovered} \right] = \frac{\int \lambda e^{-\lambda tp} dG(\lambda)}{\int (1 - e^{-\lambda tp}) dG(\lambda)}$$

$$= \frac{E[\Lambda(t) \mid \text{data}]}{c \int (1 - e^{-\lambda tp}) dG(\lambda)}$$

and as

$$R \approx c \int (1 - e^{-\lambda tp}) dG(\lambda)$$

$$E \left[\frac{\lambda e^{-\lambda tp}}{1 - e^{-\lambda tp}} \mid \text{discovered} \right] \approx \frac{1}{R} \sum_{i=1}^R \frac{\lambda_i e^{-\lambda_i tp}}{1 - e^{-\lambda_i tp}}$$

we see that the Bayes estimator can be estimated from

$$E[\Lambda(t) \mid \text{data}] \stackrel{\text{est}}{=} \sum_{i=1}^R \frac{\lambda_i e^{-\lambda_i tp}}{1 - e^{-\lambda_i tp}}$$

which is the same estimate given in Section 2.

5. ESTIMATING A FUTURE FAILURE RATE

Using the same notation as in Section 4, suppose we are now interested in estimating at time t what the failure rate would be at time $t + s$ if testing is continued for an additional s time units. As in Section 4, the Bayes estimate of $\Lambda(t + s)$ is, when c and G are known,

$$E[\Lambda(t + s) \mid \text{data to } t] = c \int \lambda e^{-\lambda p(t+s)} dG(\lambda) .$$

Now suppose c and G are unknown. We can estimate c as before from

$$c \stackrel{\text{est}}{=} \frac{R}{\int (1 - e^{-\lambda t p}) dG(\lambda)}$$

where R is the number of bugs discovered by t . Also as

$$E \left[\frac{\lambda e^{-\lambda p(t+s)}}{1 - e^{-\lambda p t}} \mid \text{discovered by } t \right] = \frac{\int \lambda e^{-\lambda p(t+s)} dG(\lambda)}{\int (1 - e^{-\lambda p t}) dG(\lambda)}$$

we see that

$$(3) \quad E[\Lambda(t + s) \mid \text{data to } t] \stackrel{\text{est}}{=} \sum_{i=1}^R \frac{\Lambda_i e^{-\Lambda_i p(t+s)}}{1 - e^{-\Lambda_i p t}}$$

where $\Lambda_1, \dots, \Lambda_R$ are the discovered failure rates by t .

If we now forget the Bayesian scenario that led to the above estimator and consider it from a more classical approach, we obtain that

$$\begin{aligned}
E \left[\sum_{i=1}^R \frac{\lambda_i e^{-\lambda_i p(t+s)}}{1 - e^{-\lambda_i p t}} \right] &= E \left[\sum_i \frac{\lambda_i e^{-\lambda_i p(t+s)}}{1 - e^{-\lambda_i p t}} (1 - \psi_i(t)) \right] \\
&= \sum_i \lambda_i e^{-\lambda_i p(t+s)} = E[\Lambda(t+s)] .
\end{aligned}$$

Also,

$$\begin{aligned}
&E \left[\left(\sum_{i=1}^R \frac{\lambda_i e^{-\lambda_i p(t+s)}}{1 - e^{-\lambda_i p t}} - \Lambda(t+s) \right)^2 \right] \\
&= \text{Var} \left(\sum_i \frac{\lambda_i e^{-\lambda_i p(t+s)}}{1 - e^{-\lambda_i p t}} \psi_i(t) + \sum_i \lambda_i \psi_i(t+s) \right) \\
&= \sum_i \frac{\lambda_i^2 e^{-2\lambda_i p(t+s)}}{1 - e^{-\lambda_i p t}} e^{-\lambda_i p t} + \sum_i \lambda_i^2 e^{-\lambda_i p(t+s)} (1 - e^{-\lambda_i p(t+s)}) \\
&\quad + 2 \sum_i \lambda_i^2 e^{-2\lambda_i p(t+s)} \\
&= \sum_i \lambda_i^2 e^{-\lambda_i p(t+s)} \left[1 + \frac{e^{-\lambda_i p(t+s)}}{1 - e^{-\lambda_i p t}} \right] .
\end{aligned}$$

The above estimate presupposes that a bug's failure rate becomes known when the bug is discovered. If this is not the case, then we can still use the data obtained by t to estimate $\Lambda(t+s)$. One approach is to note that

$$\frac{N_i(t)}{t} \approx \lambda_i$$

where $N_i(t)$ is the number of detected errors caused by bug i . Hence, using (3) we can estimate $\Lambda(t+s)$ by

$$\begin{aligned}
 \Lambda(t+s) &\stackrel{\text{est}}{=} \frac{1}{t} \sum_{i=1}^{\infty} \frac{N_i(t) e^{-N_i(t)p(t+s)/t}}{1 - e^{-N_i(t)p}} \\
 (4) \qquad &= \frac{1}{t} \sum_{i=1}^{\infty} M_i(t) \frac{ie^{-ip(1+s/t)}}{1 - e^{-ip}}.
 \end{aligned}$$

A second approach to estimating $\Lambda(t+s)$ is to note the following:

$$\begin{aligned}
 E[\Lambda(t+s)] &= \sum_i \lambda_i e^{-\lambda_i p(t+s)} \\
 &= \sum_i \lambda_i e^{-\lambda_i p t} \left(1 - \lambda_i p s + \frac{(\lambda_i p s)^2}{2} - \frac{(\lambda_i p s)^3}{3!} + \dots \right) \\
 &= \frac{1}{p} E \left[\sum_{i=1}^{\infty} \frac{M_i(t)}{t^i} i s^{i-1} (-1)^{i+1} \right].
 \end{aligned}$$

The last equality following since

$$E[M_j(t)] = \sum_i e^{-\lambda_i p t} \frac{(\lambda_i p t)^j}{j!}.$$

Hence the above suggests the possible estimator

$$(5) \qquad \Lambda(t+s) \stackrel{\text{est}}{=} \frac{1}{p} \sum_{i=1}^{\infty} \frac{M_i(t)}{t^i} i s^{i-1} (-1)^{i+1}.$$

Though we intuitively favor (4), numerical tests are needed to see whether (4) or (5) yields the better estimates. Of course, s should not be too large in relation to t for either estimate to be very effective.

We can also use the above to estimate the expected number of new bugs discovered in $(t, t+s)$. As this quantity is equal to $p E \left[\int_0^s \Lambda(t+y) dy \right]$, it follows from (3) that we can estimate this quantity by

$$\sum_{i=1}^R \int_0^S p \frac{\Lambda_i e^{-\Lambda_i p(t+s)}}{1 - e^{-\Lambda_i p t}} dy = \sum_{i=1}^R \frac{(1 - e^{-\Lambda_i p s}) e^{-\Lambda_i p t}}{1 - e^{-\Lambda_i p t}} .$$

When the failure rates do not become known when a bug is detected, we can estimate the expected number of bugs that will be discovered between t and $t + s$ by

$$\sum_i M_i(t) \frac{e^{-ip}}{1 - e^{-ip}} (1 - e^{-ips/t}) .$$

Remarks:

- (1) The results in this section can be used to devise an easily implemented stopping rule for testing. One could test for a time t and then based on the observed data choose an additional time testing time s such that the estimated error rate at s would be acceptable. One can then reevaluate this after testing for the additional time to determine whether to stop or to continue for an additional time indicated by the above.

REFERENCES

- [1] Diaconis, P., Unpublished notes on Statistical Decision Theory.
- [2] Efron, B. and R. Tibshirani, "How Many Words Did Shakespeare Know," *Biometrika*, 63, 435-447, 1976.
- [3] Goel, A. L. and K. Okumoto, "Time-Dependent Error-Detection Rate Model for Software Reliability and Other Performance Measures," *IEEE Trans. Reliability*, R-28, 206-211, 1979.
- [4] Jelinski, Z. and P. B. Moranda, "Software Reliability Research, in STATISTICAL COMPUTER PERFORMANCE EVALUATION (W. Freiberger, ed.), 485-502, Academic Press, New York, 1972.
- [5] Littlewood, B., "A Critique of the Jelinski-Moranda Model for Software Reliability," *Proc. Ann. Rel. and Maintainability Symp.*, 357-364, 1981.
- [6] Littlewood, B. and J. L. Verrall, "A Bayesian Reliability Growth Model for Computer Software," *Record IEEE Symp. Comp. Software Rel.*, 70-77, IEEE, New York, 1973.
- [7] Ramamoorthy, C. V. and F. B. Bastini, "Software Reliability-Status and Perspectives," *IEEE Transactions on Software Engineering*, Vol. SE-8, No. 4, July 1982.
- [8] Robbins, H., "Estimating the Probabilities of Unobserved Events," *Ann. of Math. Stat.*, 39, 256-257, 1968.

END

FILMED

6-85

DTIC